| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/586,916 | 07/24/2006 | Sander R. Kooijmans | US040072US | 8244 |

24737        7590        11/25/2008
PHILIPS INTELLECTUAL PROPERTY & STANDARDS
P.O. BOX 3001
BRIARCLIFF MANOR, NY 10510

| EXAMINER |
|---|
| POLLOCK, GREGORY A |

| ART UNIT | PAPER NUMBER |
|---|---|
| 3695 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 11/25/2008 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

## DETAILED ACTION

1.     This action is responsive to claims filed 08/20/2008 and Applicant's request for

reconsideration of application 10/586916 filed 08/20/2008.

The amendment contains amended claims 1, 3, and 4.

The amendment contains new claims 19-21.

Claims 2 and 5 – 18 have been canceled.


### *Claim Rejections - 35 USC § 103*

1.     The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set
forth in section 102 of this title, if the differences between the subject matter sought to be patented and
the prior art are such that the subject matter as a whole would have been obvious at the time the
invention was made to a person having ordinary skill in the art to which said subject matter pertains.
Patentability shall not be negatived by the manner in which the invention was made.


2.     Claims 1, 3, 4 and 19-21 are rejected under 35 U.S.C. 103(a) as being

unpatentable over Bacon et al. (U.S. Patent No. 6212278) in view of O'Neill (U.S.

Patent No. 6832373)


**As per claim 1,** Bacon et al. teaches **a method for updating software for use
in operation of a receiver device** (code on a subscriber terminal is modified by
downloading new code [Abstract, lines 1-4], where a subscriber terminal is more
commonly known as a set top terminal [column 1, lines 12-14])**, said receiver
device including a loader** (boot program [Abstract, lines 4-5])**, the method
comprising the steps of:**

**implementing a first and second modes** (an off mode [column 15, lines 19-22]
and an "interactive session" [column 15, lines 9-12]) **on the receiver device (50)**
(Figure 1, elements 40, 44, and 48 shows various means for receiving
downloaded data from the headend source [column 4, line 43 – column 5, line

39]) **such that at least a first mode is a normal operating mode for the receiver device** ("interactive session" [column 15, lines 9-12]) **and a second mode (10) is a standby mode** (an off mode [column 15, lines 19-22]);

**controlling the receiver device such that the receiver device loader in one of the standby mode or normal operating mode searches for software updates** (software update are transmitted by the headend at anytime [column 4, line 43 – column 5, line 39] and can be received by the subscriber in either mode for downloading as evident by the necessity of the convenience flag [column 15, lines 9-18]); **transfers the available software update to the receiver device and stores the same in memory** (the MCC controls input of data from the headend and buffers it in volatile memory area. [column 2, lines 11-27]);

**and installing (22) transferred software updates into the programmable device** (Figure 9 shows the data being transfers from volatile memory DRAM (dual port ram) to non-volatile memory (FLASH memory) [column 1, lines 47-48] and [column 13, lines 3-13])

Bacon et al. does not teach **that installation of software takes place when the receiver device is changed into the other of the standby or normal operating mode.**

O'Neill teaches a method where **the loader indicates (44) to the other of the modes with the receiver device an occurrence of an available software update** (indications that the software has been successfully stored downloaded into a status table [column 32, lines 7-11]) **and installation of software takes place when the receiver device is changed into the other of the standby or normal operating mode** (A download agent retrieves available software updates [column 28, lines 47-49] which is loaded into a volatile memory [column 31, lines 65-67] and places and indication that the software has been successfully stored downloaded into a status table [column 32, lines 7-11]. An upload agent installs the downloaded software [column 28, lines 1-10] to non-volatile flash memory [column 31, line 67 – column 32, line 3] when a mode change is detected [column 31, lines 55-63]).

It would have been obvious to one skilled in the art at the time of the invention to have used the download agent, upload agent, and status table as found in the invention of O'Neill within the invention of Bacon et al. to achieve the claimed invention. Bacon et. al. contains all of the required elements (volatile RAM, non-volatile RAM, flag register, and a loader) such that use with the invention of O'Neill can readily be achieved. One would be motivated to combine the inventions because O'Neill provides a fault tolerant process comprises a series of checks or validations that advantageously protect the existing code version that is used by the client device, to avoid corruption during execution of the

instructions contained in the update package. Furthermore, the fault tolerant process insures that the information contained in the updated code accurately reflects the desired information to insure proper device operation after the update. Advantageously, the above-mentioned fault tolerant process is used to protect the operational software or firmware code against unforeseen failure, such as a power failure or data corruption. When a power failure occurs in the installation process, the update agent is able to determine the banks that have been updated, the next bank to be updated, and when the update installation is complete. Furthermore, the fault tolerant process efficiently and reliably protects the bank update installation and conversion process against hardware or software interruptions, power failures, data corruption, and/or other catastrophic encounters beyond the control of the user.

**As per claim 3,** the rejection of claim 1 has been addressed.
Bacon et al. teaches **a method wherein the steps of searching (16) and transferring (28) are done in the standby mode** (software updates are transmitted by the headend at anytime [column 4, line 43 – column 5, line 39] and can be received by the subscriber in either mode for downloading as evident by the necessity of the convenience flag [column 15, lines 9-18]).

**As per claim 4,** the rejection of claim 1 has been addressed.
Bacon et al. teaches **a method wherein the steps of searching (16) and transferring (28) are done in the normal operating mode** (software updates are transmitted by the headend at anytime [column 4, line 43 – column 5, line 39] and can be received by the subscriber in either mode for downloading as evident by the necessity of the convenience flag [column 15, lines 9-18]).

**As per claim 19,** the rejection of claim 1 has been addressed.
Bacon et al. teaches **a method wherein the software update is initially installed into volatile memory in the receiver device and then subsequently transferred into a non-volatile memory** (Figure 9 shows the data being transfers from volatile memory DRAM (dual port ram) to non-volatile memory (FLASH memory) [column 1, lines 47-48] and [column 13, lines 3-13]).

**As per claim 20,** the rejection of claim 1 has been addressed.
Bacon et al. teaches **a method wherein the standby mode does not function simultaneously with the normal operating mode** (off mode [column 15, lines 19-22]).

**As per claim 21,** the rejection of claim 1 has been addressed.
Bacon et al. does not teach **a method wherein the software update is initially installed into volatile memory in standby mode and then subsequently transferred into a non-volatile memory after re-entering the normal operating mode.**

O'Neill teaches a method where **a method wherein the software update is initially installed into volatile memory in standby mode and then subsequently transferred into a non-volatile memory after re-entering the normal operating mode** (A download agent retrieves available software updates [column 28, lines 47-49] which is loaded into a volatile memory [column 31, lines 65-67] and places and indication that the software has been successfully stored downloaded into a status table [column 32, lines 7-11]. An upload agent installs the downloaded software [column 28, lines 1-10] to non-volatile flash memory [column 31, line 67 – column 32, line 3] when a mode change is detected [column 31, lines 55-63]. Here, it is noted that the temporal ordering of the download to volatile memory and transfer to non-volatile memory as related to the modes is not stated within the claim.).

It would have been obvious to one skilled in the art at the time of the invention to have used the download agent, upload agent, and status table as found in the invention of O'Neill within the invention of Bacon et al. to achieve the claimed invention. Bacon et. al. contains all of the required elements (volatile RAM, non-volatile RAM, flag register, and a loader) such that use with the invention of O'Neill can readily be achieved. One would be motivated to combine the inventions because O'Neill provides a fault tolerant process comprises a series of checks or validations that advantageously protect the existing code version that is used by the client device, to avoid corruption during execution of the instructions contained in the update package. Furthermore, the fault tolerant process insures that the information contained in the updated code accurately reflects the desired information to insure proper device operation after the update. Advantageously, the above-mentioned fault tolerant process is used to protect the operational software or firmware code against unforeseen failure, such as a power failure or data corruption. When a power failure occurs in the installation process, the update agent is able to determine the banks that have been updated, the next bank to be updated, and when the update installation is complete. Furthermore, the fault tolerant process efficiently and reliably protects the bank update installation and conversion process against hardware or software interruptions, power failures, data corruption, and/or other catastrophic encounters beyond the control of the user.

### Response to Arguments

34.　Applicant's arguments with respect to claims 1, 3, 4, and 19-21 have been

considered but are moot in view of the new ground(s) of rejection necessitated by

applicant's claim amendments The rejection above serves as the examiners response to the applicant's arguments.

### Conclusion

3.     Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

4.     The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Aoyama (US Patent: 7398331) – teaches A peripheral apparatus reserves and stores, on a nonvolatile storage medium, a first storage area which stores a control program for implementing a function of the peripheral apparatus, and a second storage area which stores a rewrite program for rewriting the control program. The firmware is formed from at least the control program and the rewrite program. When selectively switching and controlling a first mode for updating the firmware of the rewrite program, and a second mode for executing the control program, the first mode receives updating data of the control program from an external apparatus, rewrites the control program stored in the first storage area to updating data, using the rewrite program, determines that a rewrite operation is normally ended, and performs the rewrite operation again when the rewrite operation is not normally ended.

- Chambers, II et al. (US Patent: 6779176) – teaches a method of implementing program updates in an electronic system includes requesting the system to accept a program update and loading replacement programs for execution. The method includes using an operating system to determine when programs running on an electronic system have finished an execution cycle and then commencing the replacement process.

- Meyer et al. (US Patent: 7016944) – teaches a system and method that monitors upgrade availability for computer information on a user's computer and allows the user to determine which of the available upgrades will be downloaded to the user's computer and installed. The upgrade availability for computer information on the user's computer is monitored in the background, without user-intervention when the user connects to a network, such as the Internet. If any such upgrades are available, a flag is set to notify the user of such upgrades. The user is notified of any available upgrades when computer information is accessed for which an upgrade is available, and given a choice of whether or not to download the available upgrade(s). The downloaded upgrade(s) may then be installed by an installation means. In this manner, the present invention allows for the user to download only those upgrades desired by the user.

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed

within TWO MONTHS of the mailing date of this final action and the advisory

action is not mailed until after the end of the THREE-MONTH shortened statutory

period, then the shortened statutory period will expire on the date the advisory

action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be

calculated from the mailing date of the advisory action. In no event, however, will

the statutory period for reply expire later than SIX MONTHS from the date of this

final action.

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Gregory Pollock whose telephone number is 571

270-1465. The examiner can normally be reached on 7:30 AM - 4 PM, Mon-Fri

Eastern Time.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Jay Kramer can be reached on 571 272-6783. The fax phone

number for the organization where this application or proceeding is assigned is

571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR

only. For more information about the PAIR system, see http://pair-

direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-

free). If you would like assistance from a USPTO Customer Service

Representative or access to the automated information system, call 800-786-

9199 (IN USA OR CANADA) or 571-272-1000.

GAP

11/14/2008

/Gregory Pollock/
Examiner, Art Unit 3695

Gregory A. Pollock

/Lewis A. Bullock, Jr./
Supervisory Patent Examiner, Art Unit 2193